Session 42: So Your Predictive Model is Turned On. Now What?

Munich RE

So, Your Predictive Model is Turned On.

Now What?

Adnan Haque
Integrated Analytics
Munich Re

Machine learning in life insurance

Accelerate underwriting

- Eliminate evidence
- Automate decisions

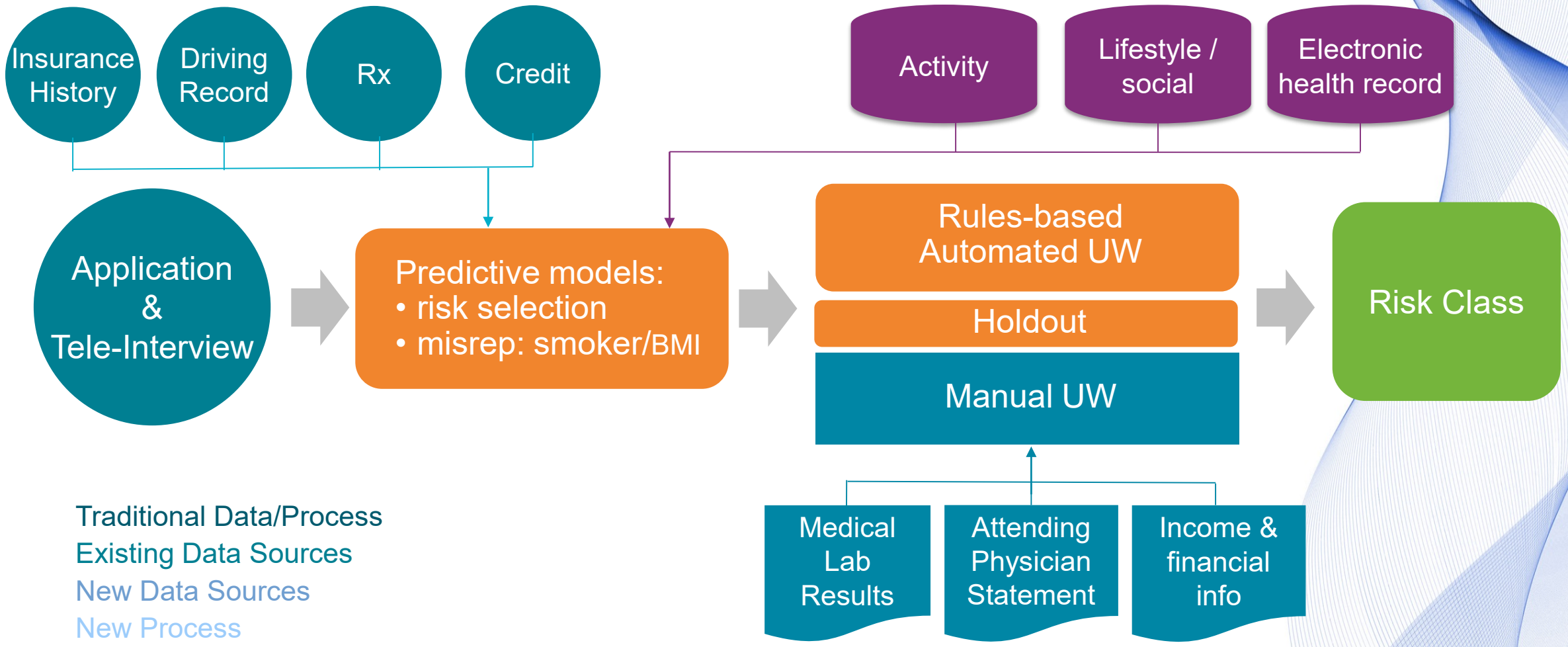
Price more accurately

- Incorporate more factors into mortality / morbidity prediction
- Provide finer segmentation or even individual pricing

Drive sales and marketing with data

- New models of IT
- Target for both marketing and risk

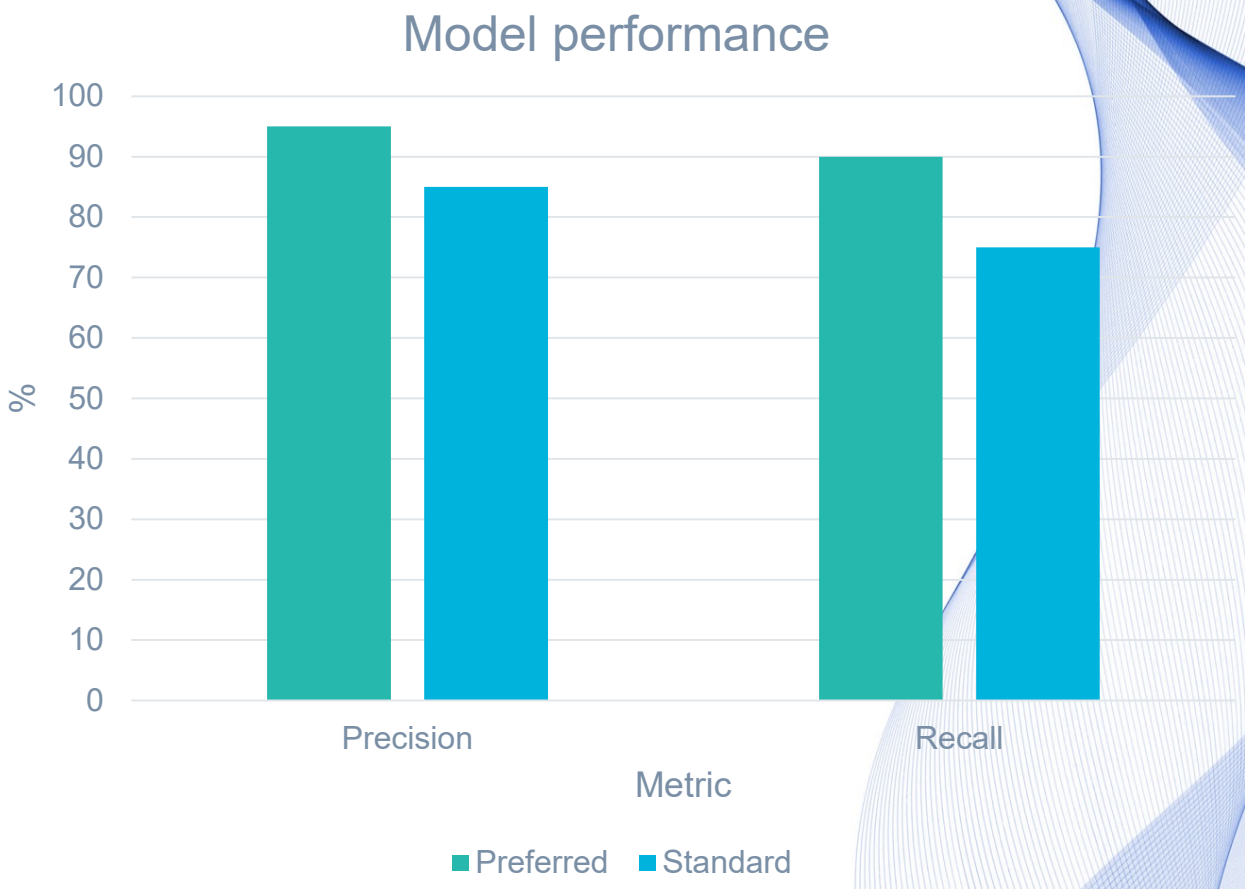
Model case study



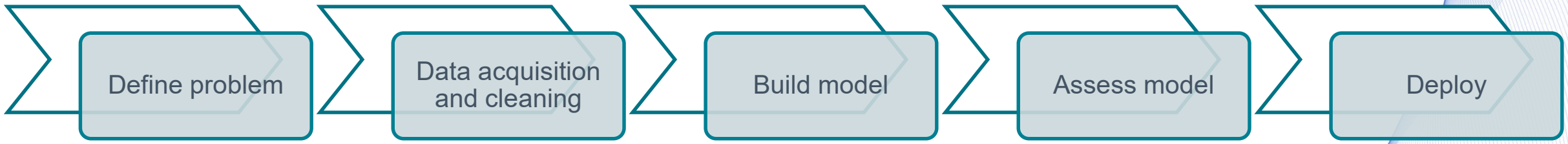
Traditional Data/Process
Existing Data Sources
New Data Sources
New Process

Risk selection model

Case	Actual UW Class (with labs/exams)	Model Predicted Probabilities: Preferred	Standard	Refer	Predicted UW Class
Case 1	Preferred	99%	1%	0%	Preferred
Case 2	Declined	50%	30%	20%	Refer
Case 3	Preferred	70%	15%	5%	Preferred
Case 4	Declined	90%	8%	2%	Preferred
Case 5	Standard	20%	75%	5%	Standard



Model lifecycle 5 years ago

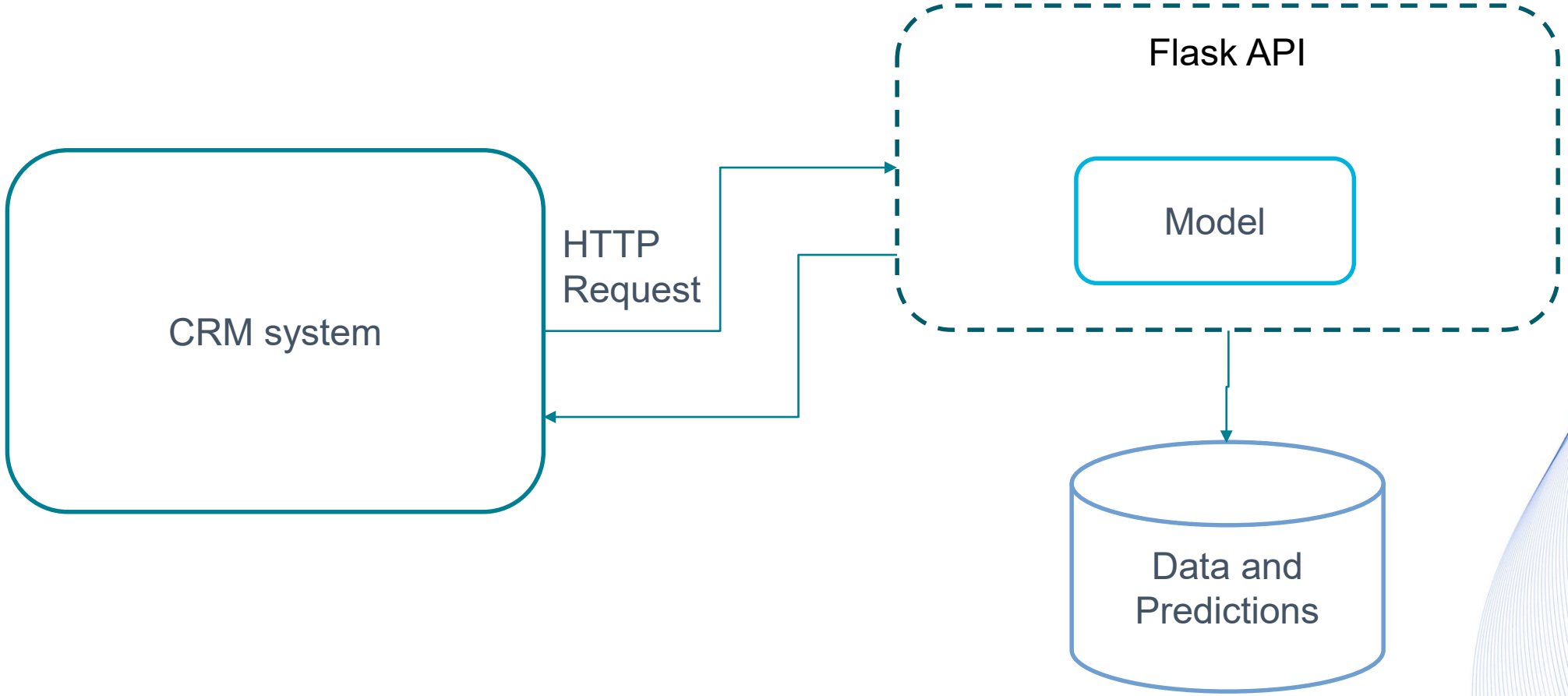


Am I done?

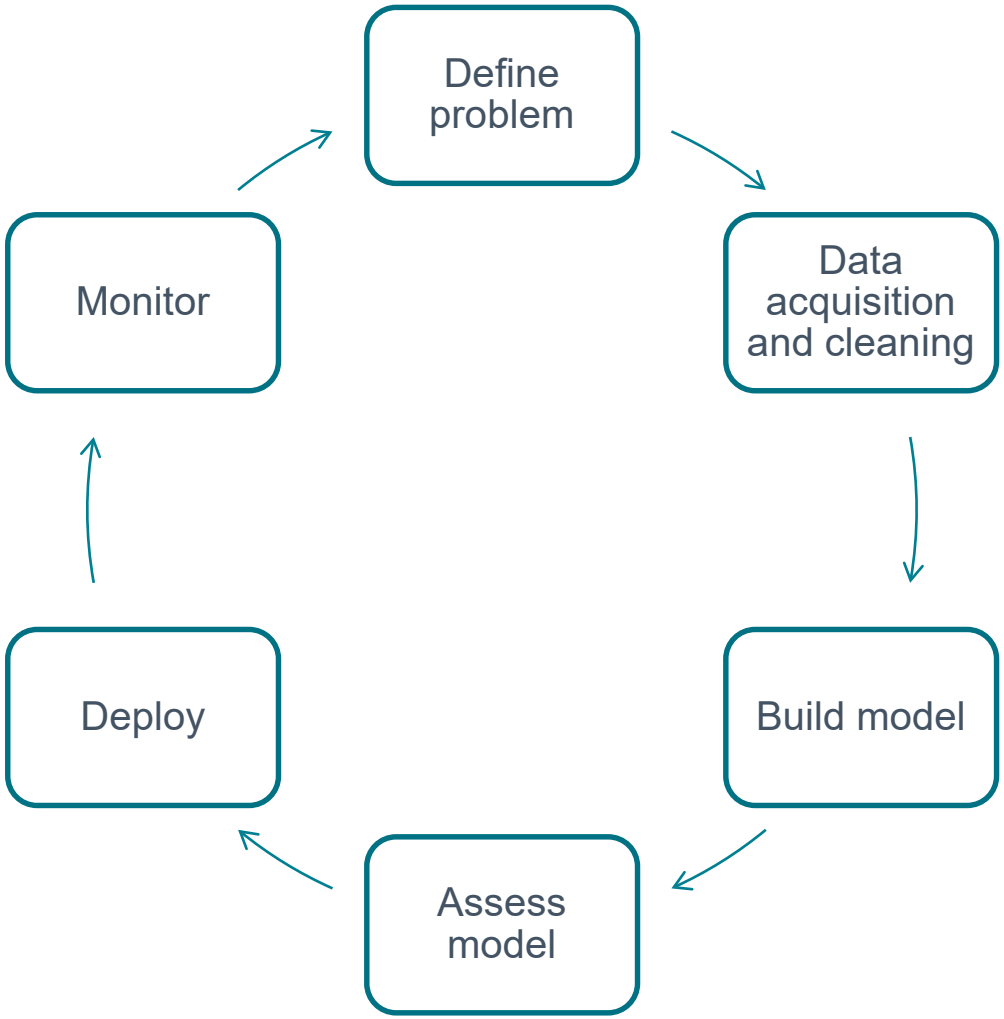
```
def train(X, y):
"""Train model."""
  clf = RandomForestClassifier()
  clf.fit(X, y)
  with open('model.pickle', 'wb') as f:
    pickle.dump(clf, f)

def predict(X):
  """Model prediction."""
  with open('model.pickle', 'rb') as f:
    clf = pickle.load(f)
  return clf.predict(X)
```

Deployment architecture



Model lifecycle today



Why model monitoring matters

Why do models go wrong?

- World changes, training data might no longer depict real world
- Model inputs might change
- Undiscovered bugs in data pipeline or model
- Model becomes worse over time

Input monitoring

- Schema validation
- Correlation checks
- Distribution checks
- Clustering
- Model driven anomaly detection

Input monitoring – schema validation

- Type checks
- Minimum and maximum ranges
- Unseen categories

Swagger Editor File ▾ Edit ▾ Generate Server ▾ Generate Client ▾ Switch back to previous editor

```yaml
swagger: "2.0"
info:
  description: "This is a sample server Petstore server.  You can find
    out more about     Swagger at [http://swagger.io](http://swagger.io)
    or on [irc.freenode.net, #swagger](http://swagger.io/irc/).      For
    this sample, you can use the api key `special-key` to test the
    authorization     filters."
  version: "1.0.0"
  title: "Swagger Petstore"
  termsOfService: "http://swagger.io/terms/"
  contact:
    email: "apiteam@swagger.io"
  license:
    name: "Apache 2.0"
    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
host: "petstore.swagger.io"
basePath: "/v2"
tags:
- name: "pet"
  description: "Everything about your Pets"
  externalDocs:
    description: "Find out more"
    url: "http://swagger.io"
- name: "store"
  description: "Access to Petstore orders"
- name: "user"
  description: "Operations about user"
  externalDocs:
    description: "Find out more about our store"
    url: "http://swagger.io"
schemes:
- "http"
paths:
  /pet:
    post:
      tags:
      - "pet"
      summary: "Add a new pet to the store"
      description: ""
```

Swagger Petstore 1.0.0

[Base URL: petstore.swagger.io/v2]

This is a sample server Petstore server. You can find out more about Swagger at http://swagger.io or on irc.freenode.net, #swagger. For this sample, you can use the api key `special-key` to test the authorization filters.

Terms of service
Contact the developer
Apache 2.0
Find out more about Swagger

Schemes
HTTP

Authorize

pet Everything about your Pets Find out more: http://swagger.io

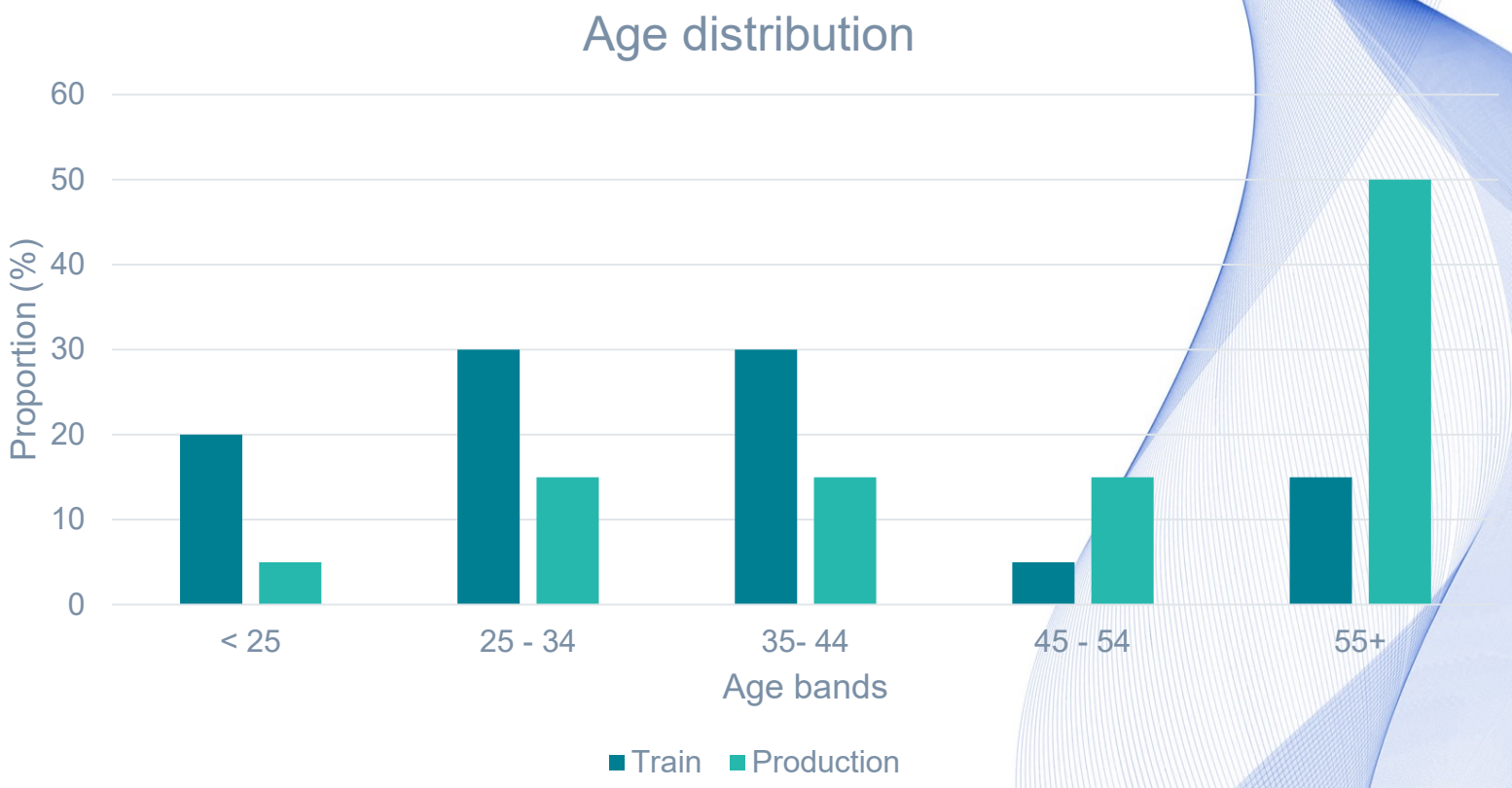
POST /pet Add a new pet to the store

PUT /pet Update an existing pet

Input monitoring – distribution

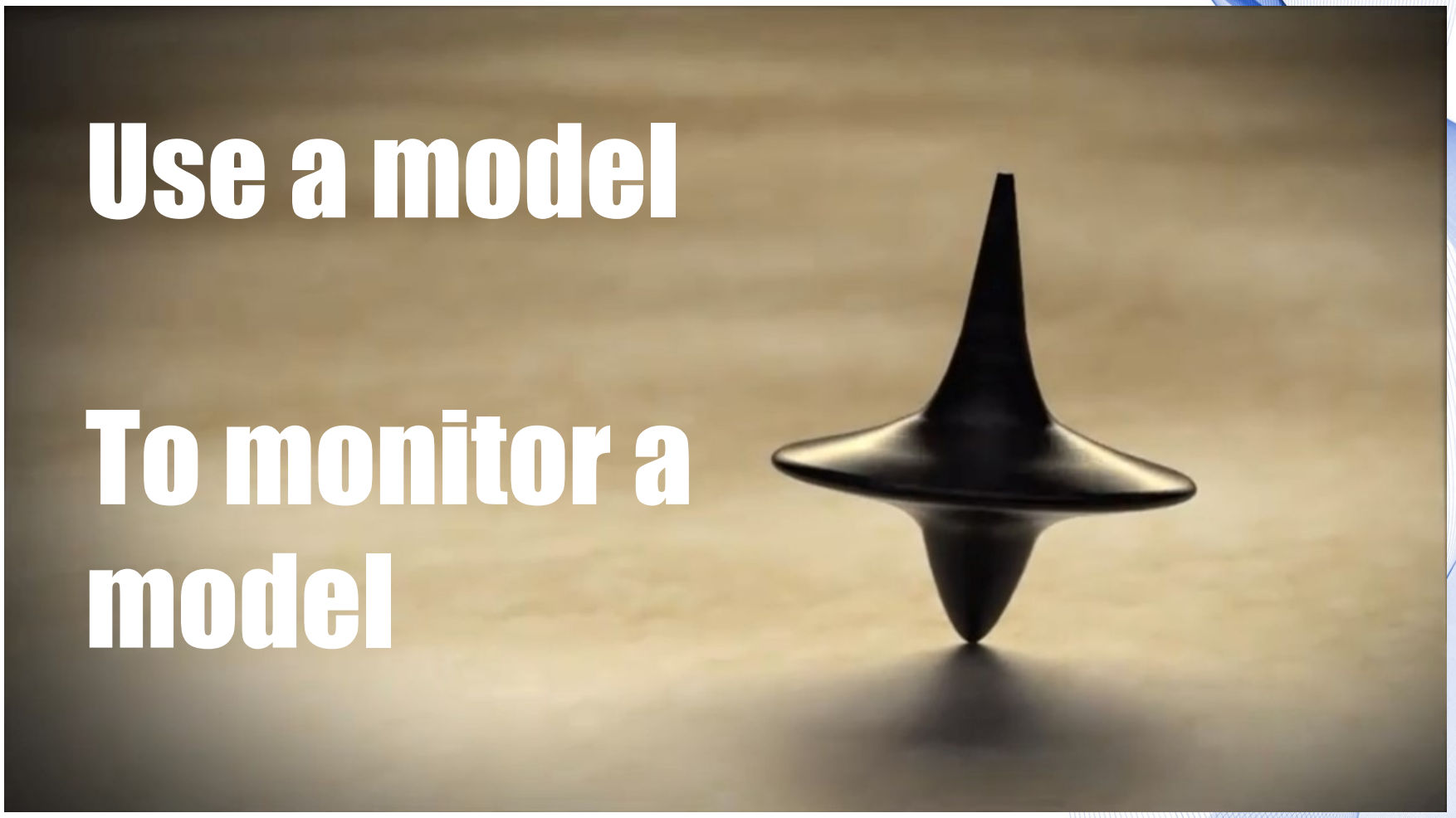
- Missingness
- Metrics
  - Kolmogorov-Smirnov test
  - Kullback-Leibler divergence
  - L-infinity distance

great-expectations library



Input monitoring – anomaly detection

- Random Cut Forests
- Generative Adversarial Networks



Output monitoring

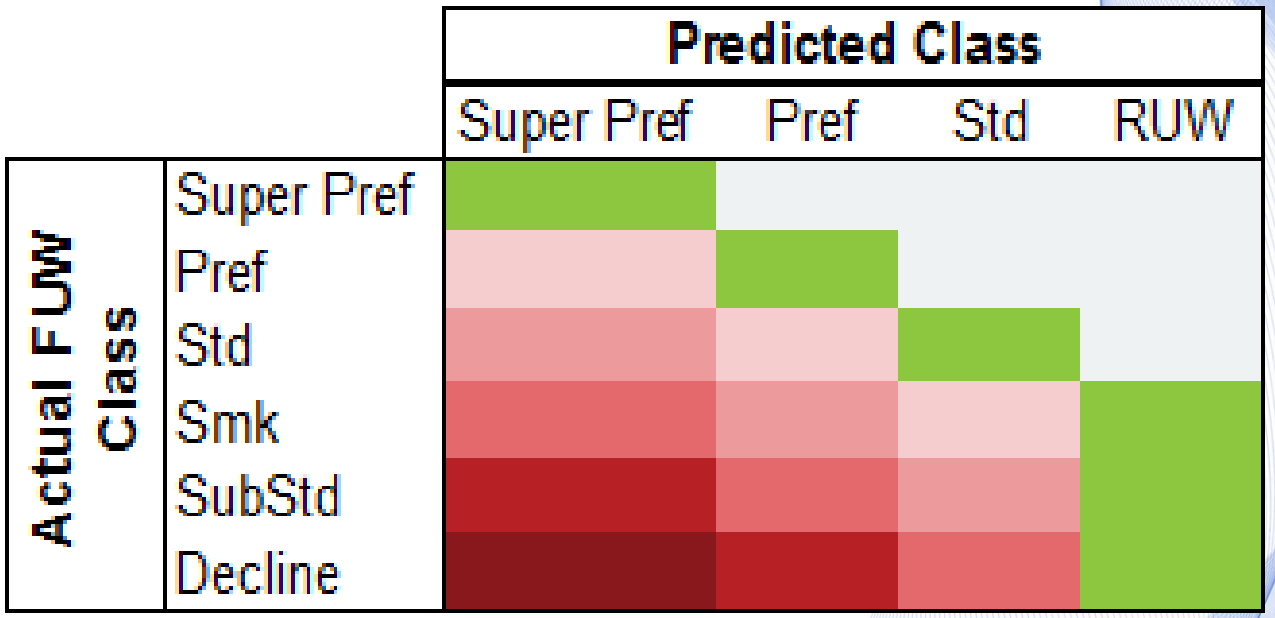
- Distribution of output
- Compare against ground truth
- Model confidence
- Compare against other model predictions

Output monitoring – ground truth

Misclassification matrix

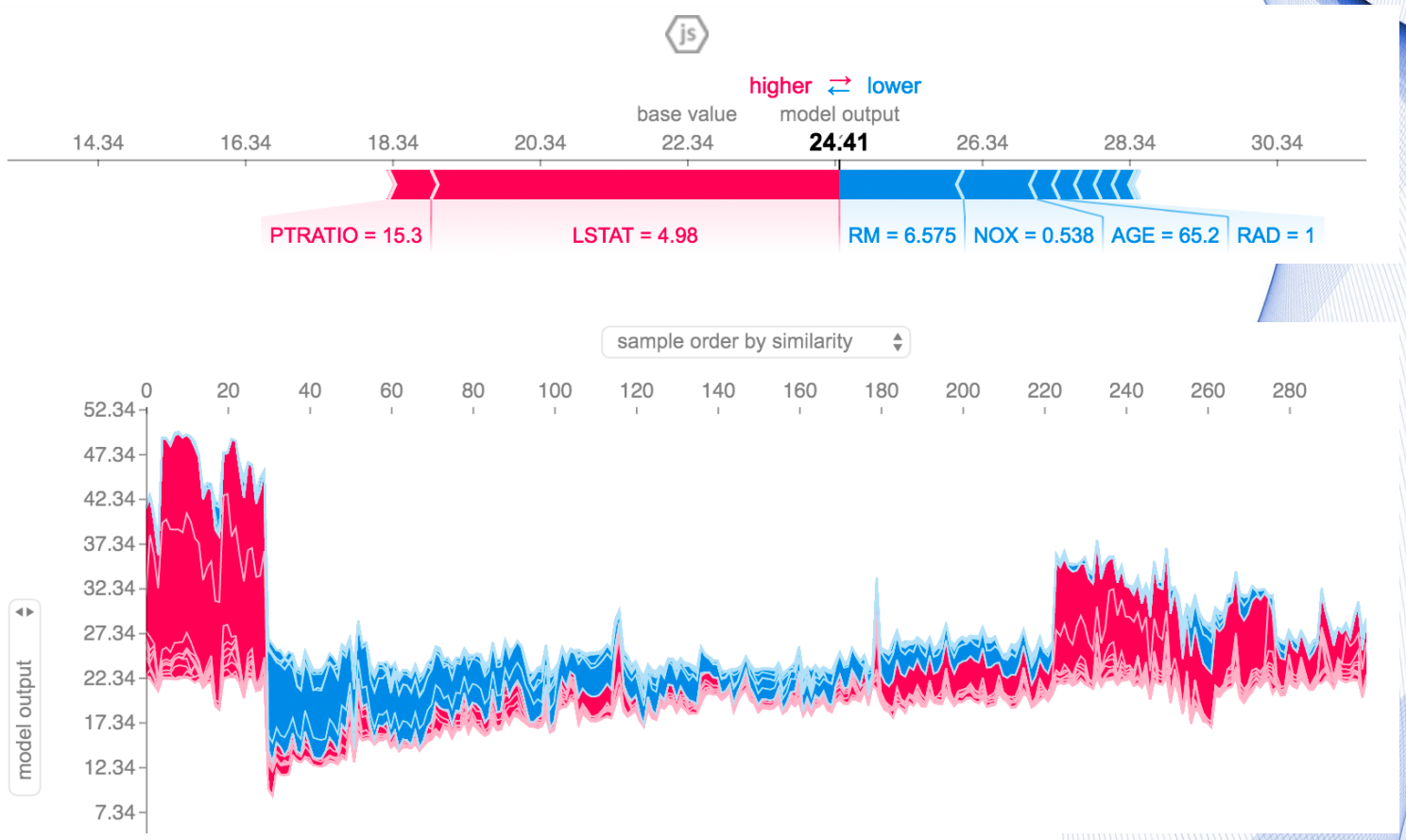
Actual UW Class	Predicted Class: Preferred	Standard	Decline
Preferred	1000	0	0
Standard	50	200	0
Decline	10	30	100

Cost matrix



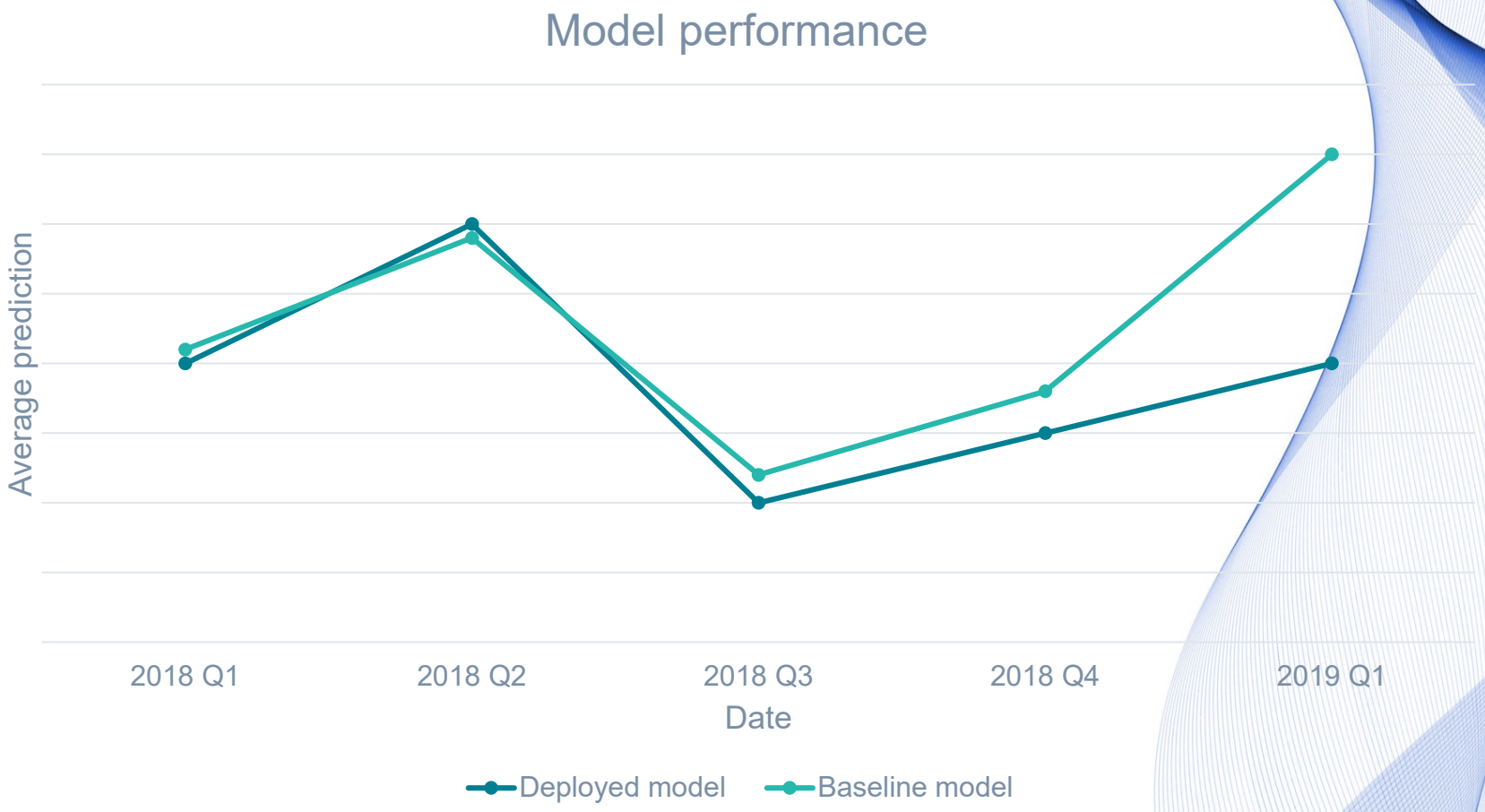
Output monitoring – feature importance

- Track aggregated feature importance over time
- Track reason codes assigned



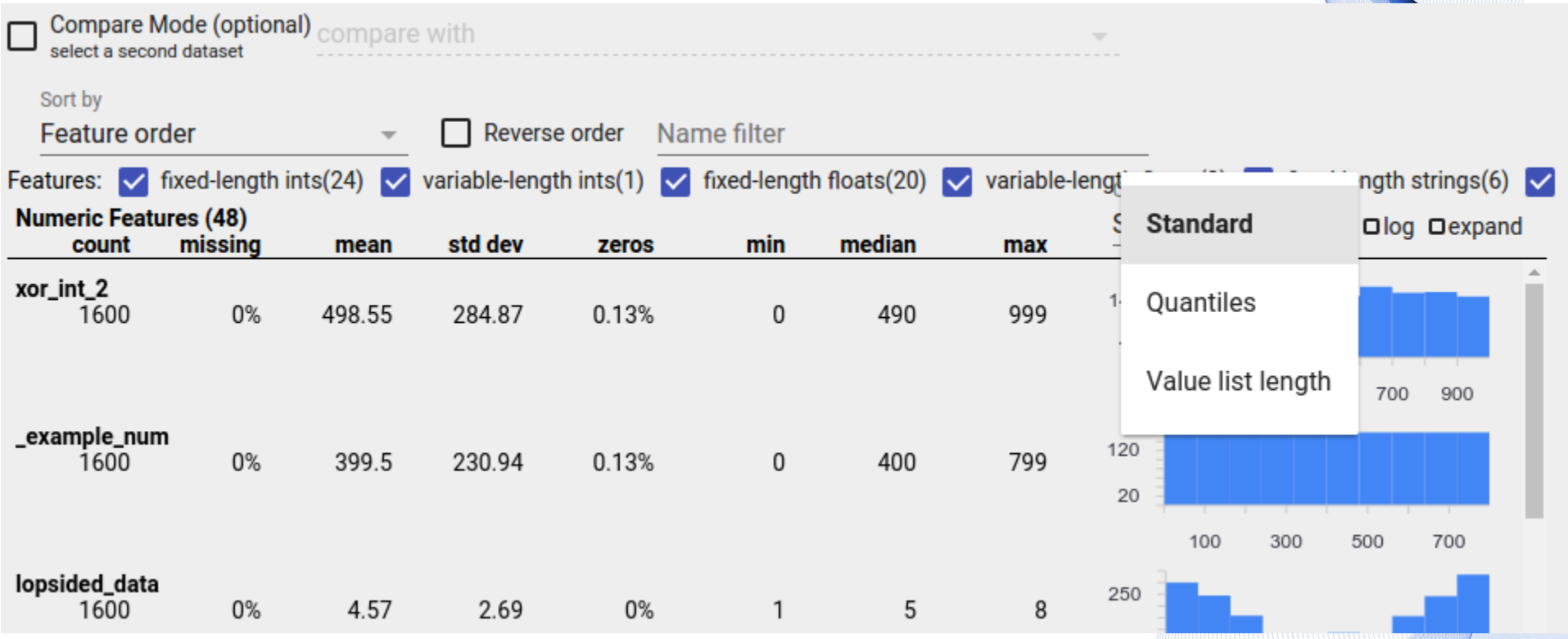
Output monitoring – baseline model

- Train and deploy a simple baseline model in conjunction with a more sophisticated deployed model



Managing alerts

- Dashboard
- Minimize the number of false positives



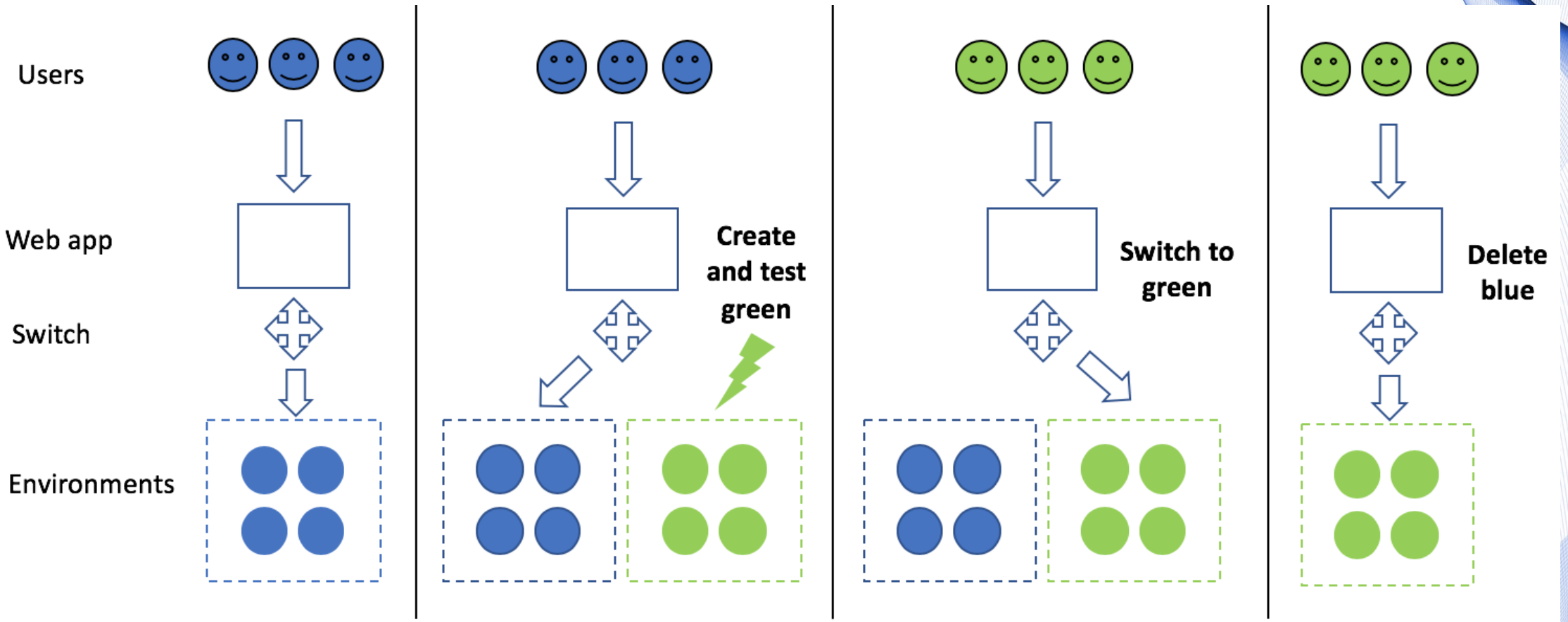
Batch learning

- Scheduled vs. trigger based
- How large should our window be?
- When do we accept our retrained model?

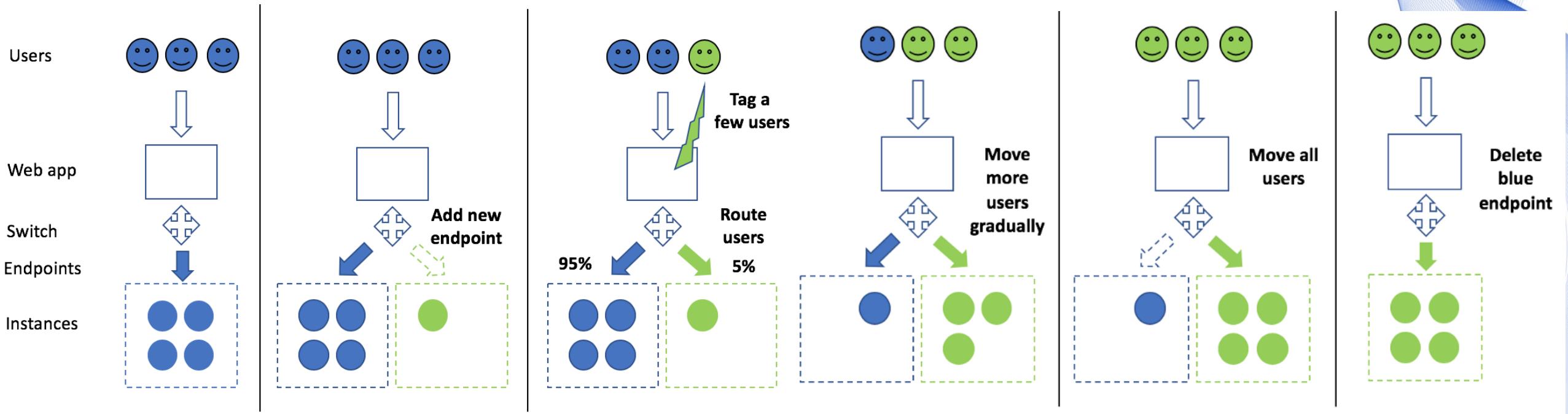
Model versioning

- Reproducibility matters!
  - All code should be in code repository (git)
  - Data and code should be tied together
  - Environment reproducibility (docker)
- Allow for seamless rollbacks

Blue-green deployment

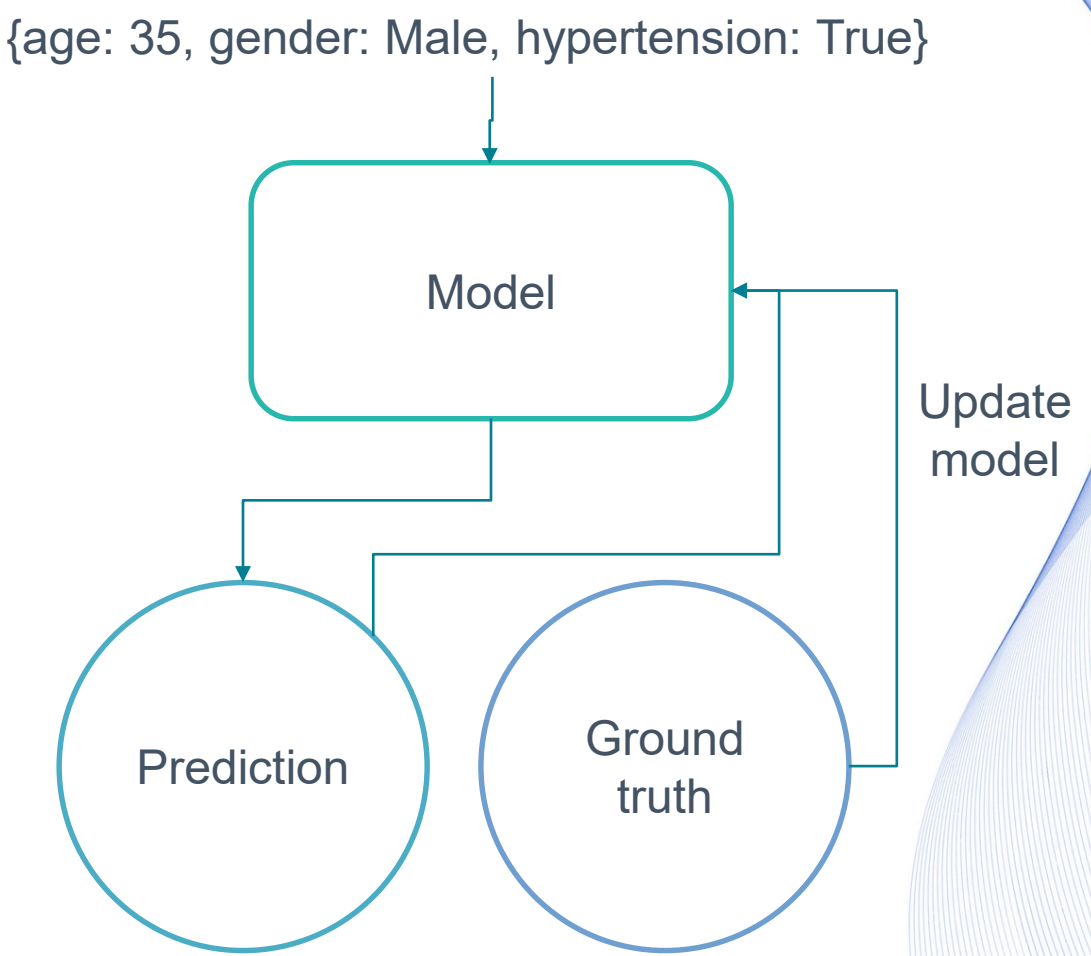


Canary testing

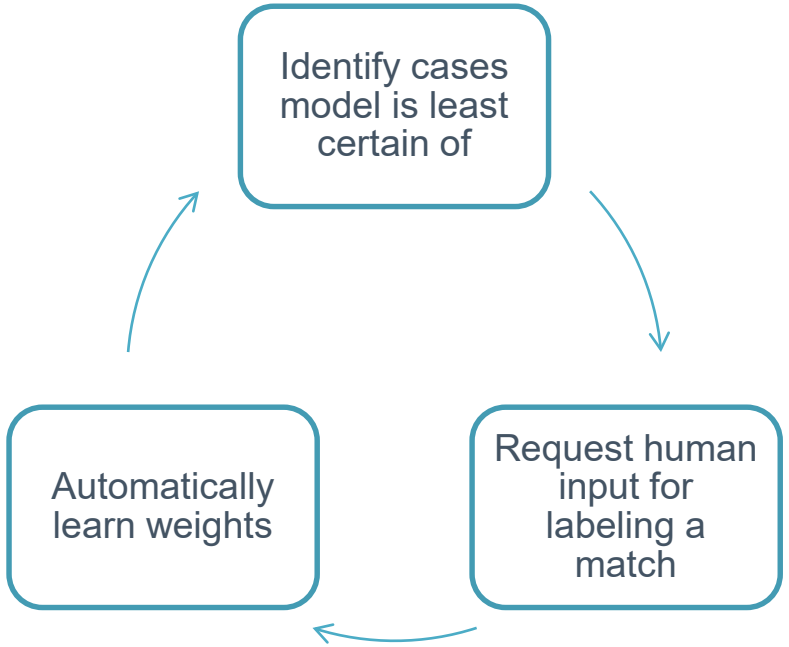


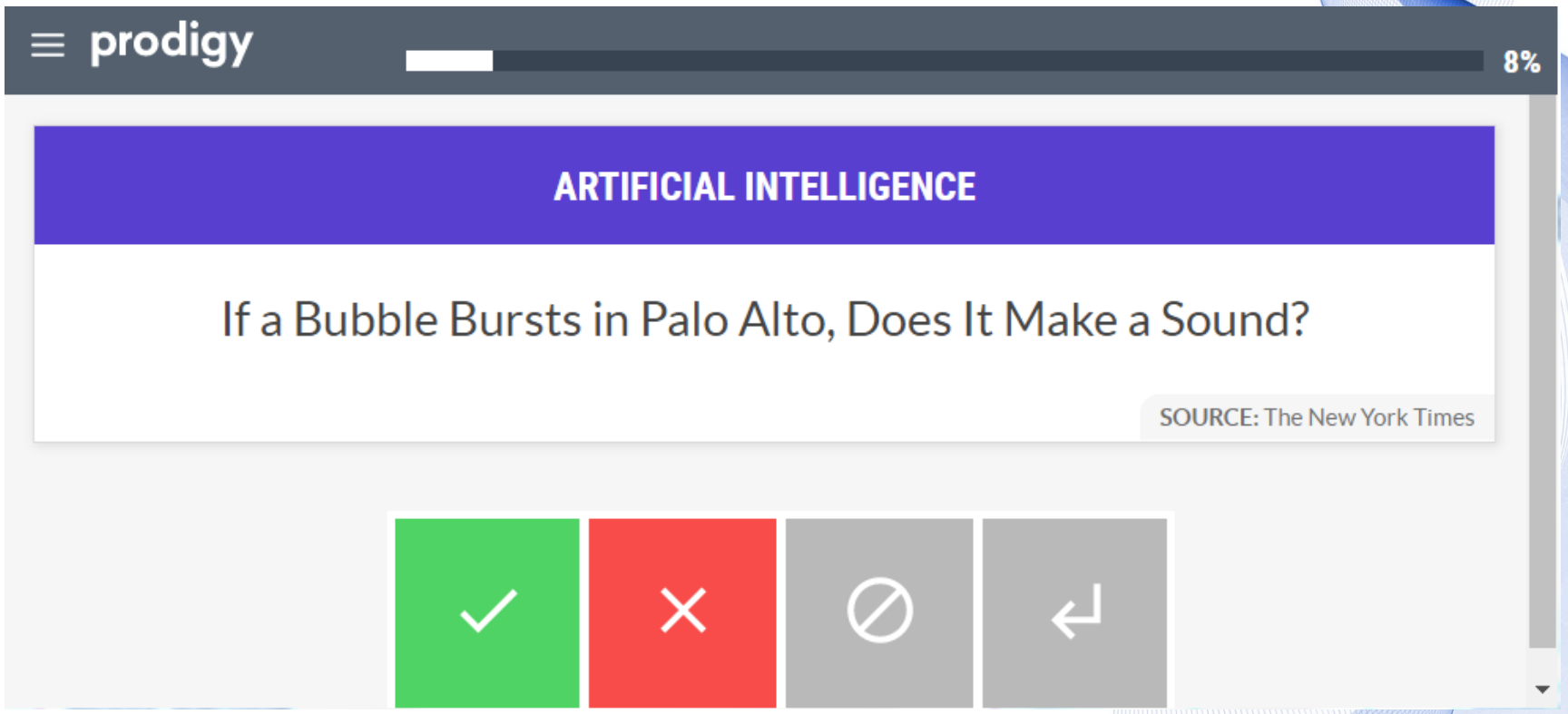
Online learning

- Leverages continuous stream of ground truth
- Adapts to emerging relationships
- Expensive to maintain
- Difficult to have tractability

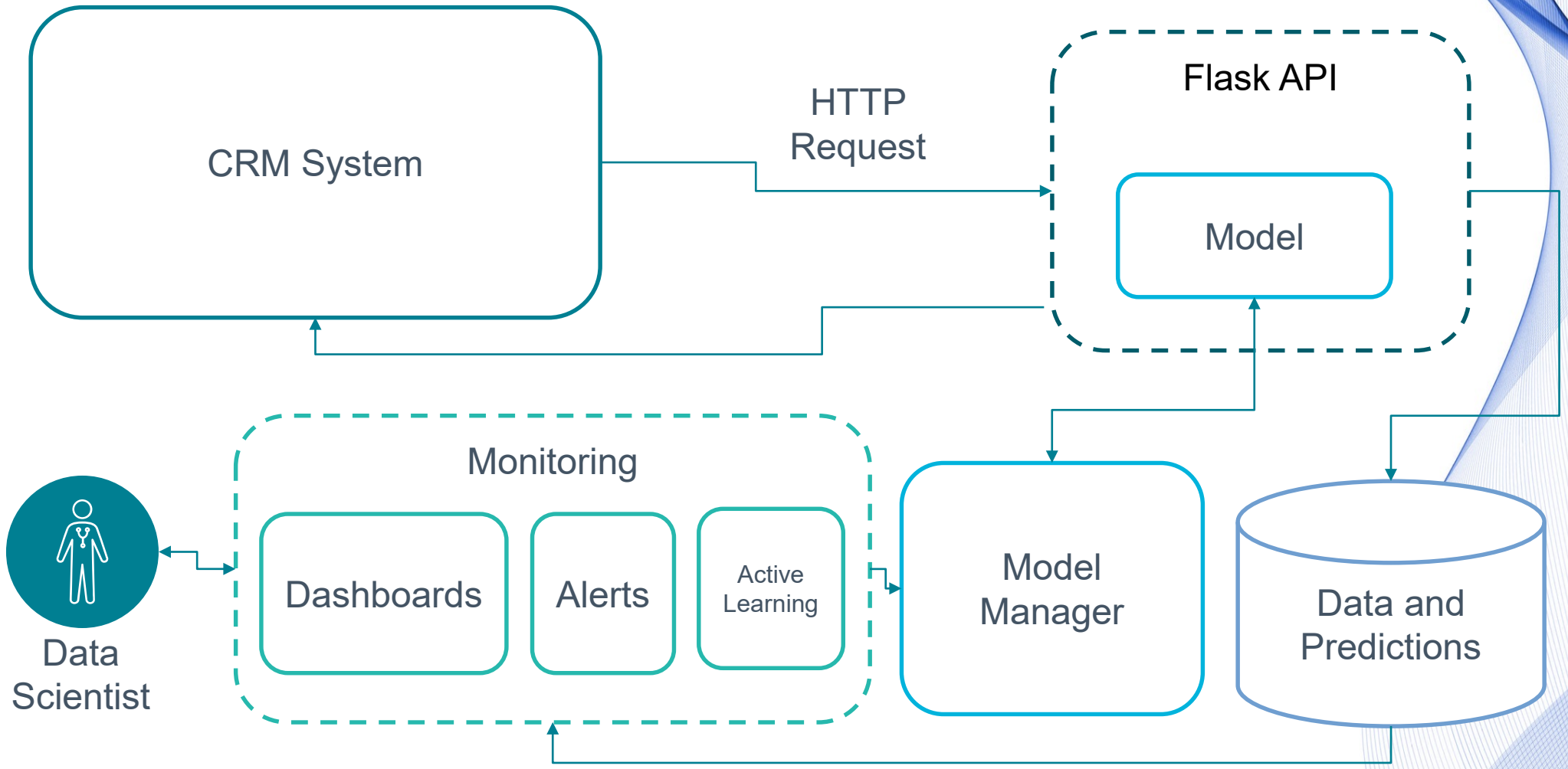


Active learning





Overall architecture



ML test score

What's your Machine Learning Test Score? A rubric for ML production systems. (https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf)

- **0 points**: More of a research project than a productionized system.
- **1-2 points**: Not totally untested, but it is worth considering the possibility of serious holes in reliability.
- **3-4 points**: There's been first pass at basic productionization, but additional investment may be needed.
- **5-6 points**: Reasonably tested, but it's possible that more of those tests and procedures may be automated.
- **7-10 points**: Strong levels of automated testing and monitoring, appropriate for mission critical systems.
- **12+ points**: Exceptional levels of automated testing and monitoring

Future reading

- Hidden Technical Debt in Machine Learning Systems (https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf)
- Operational monitoring
- MLflow (https://mlflow.org/)
- TFDV (Tensorflow Data Validation) and TFMA (Tensorflow Model Analysis) (https://www.tensorflow.org/tfx/tutorials/model_analysis/chicago_taxi)
- Racket (https://racket.readthedocs.io/en/latest/)

Thank you!